**Software Testing Report**

**For**

**Armadillo 4000**

**Version 1.1 approved**

**Prepared by Juergen Kriz, Ryan Mock, Sean Dunbar, Sunveer Sandhu, Zach Cockle**

**March 01, 2018**

Table of Contents

[1.0 INTRODUCTION 3](#_Toc510106858)

[1.1 PURPOSE 3](#_Toc510106859)

[2.0 TEST ASSESSMENT 3](#_Toc510106860)

[3.0 TEST RESULTS 3](#_Toc510106861)

[3.1 UNIT TESTING 3](#_Toc510106862)

[3.2 INTEGRATION TESTING 3](#_Toc510106863)

[3.3 SYSTEM TESTING 3](#_Toc510106864)

# 1.0 INTRODUCTION

## 1.1 PURPOSE

This Armadillo 4000 Test Report provides a summary of the results of tests performed as outlined within this document.

# 2.0 TEST ASSESSMENT

We did many small tests, attempting to break our individual functionalities in every way possible. As listed in the table below, we were quite successful in this. However, alongside the bugs/faults came various fixes and changes to accommodate these issues, and we managed to solve 95% of all the documented errors. We have also done some complete program tests, pushing every functionality, trying to crash the program. We feel we have done sufficient testing to ensure we can produce the best version of our program.

# 3.0 TEST RESULTS

There was a test after every attempted fix, with many failures, but eventually many successes as well. The overall design of the program did change slightly, as we overhauled the code to make all the pieces fit together more efficiently. This did not lessen the functionality of our program, but improved it. We also added more of a GUI than originally planned, instead of doing a very simple command line design. One major issue encountered with our program is that the API occasionally has problems calling cryptocurrency, for no explainable reason; it will simply make the call and return with no data whatsoever.

## 3.1 UNIT TESTING

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect | Summary of Defect | Closed prior to Production Release? | Comments |
| A1 | March 15 | Group | Fail | Critical | The *stSMA* & *line* variable are not setting properly, remaining uninitialized. | No | This is causing the *while* loop to not be entered, therefore the program does not run. |
| A2 | March 15 | Group | Pass | Critical | The *stSMA* & *line* variable are setting properly, initializing correctly. | Yes | The while loop now enters properly, with the API output parsing properly. |
| C1 | March 15 | Group | Fail | Major | The *currentShortTerm* & *currentLongTerm* variables are not setting, rendering the SMA rule invalid/broken. | No | Without receiving values for SMA, we cannot run this section of our program. However, if the user does not choose SMA, this can go unnoticed. |
| C2 | March 15 | Group | Fail | Major | The *Symbol* variable was not receiving input, which was causing the API to call with improper data. | No | We hardcoded the *Symbol* variable, and now we get *currentShortTerm*, but we do not get *currentLongTerm*. |
| C3 | March 15 | Group | Pass. | Major | Program was feeding the API values into the wrong spots. | Yes | Fixing the API call format allowed proper retrieval of values. |
| D1 | March 15 | Group | Fail | Trivial | The program is printing out the *stSMA* variable, despite the if condition not being matched. | No |  |
| D2 | March 15 | Group | Pass | Trivial | The program no longer prints the *stSMA* unintentionally. | Yes | We left a test *system.out.println(stSMA)*, that we forgot about. The statement has since been deleted. |
| G1 | March 19 | Group | Fail | Major | The Price class is not printing results, instead giving “String literal is not properly closed by a double-quote”. | No | The FinancialPrice class is essentially broken, but the other rules still function properly if Price isn’t used. |
| G2 | March 19 | Group | Fail | Trivial | The Price class prints results, but is taking the period character and putting it on the line below. | No | The class works fine, this is purely a visual issue. |
| G3 | March 19 | Group | Pass | Trivial | The Price class prints results, and the period is not moved to another line. | Yes | This happens because we removed the period… |
| H1 | March 19 | Group | Fail | Major | When monitoring the crypto market, the program jumps straight into our compareValues method instead of prompting the user for more input. | No | The program should ask the user if they want to enter more stocks, but instead seems to be running the program by itself. |
| H2 | March 19 | Group | Pass | Major | The program now returns to prompt the user for more input, as it should. | Yes | Upon removing an unnecessary switch, a break was taking place and skipping the rest. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major revamp of code | | | | | | | |
| J1 | March 22 | Group | Fail | Minor | When a percentage decrease is entered, the program does not add the value to the API call, making it arbitrary. | No | The positive percentage works, and this does not crash the program. It only lowers functionality. |
| J2 | March 22 | Group | Fail | Minor | The negative value is overwriting the previous value, and is also not being turned negative. | No | As stated in J1. |
| J3 | March 22 | Group | Pass | Minor | The program now adds the values into the relevant variables, allowing the API to call properly. | Yes | The splitting of one variable into two allows correct placement of values. |
| K1 | March 22 | Group | Fail | Trivial | The program is displaying “Enter the ticker symbol:” twice, despite only requiring one printout. | No | This doesn’t affect functionality, only a visual bug. |
| K2 | March 22 | Group | Pass | Trivial | The program was looping back onto a single *println* statement. | Yes | Adjusted the loop to not get stuck on the *println* statement. |
| L1 | March 22 | Group | Fail | Major | The program is halting when attempting the RSI API call, shutting down this component. | No | The rest of the program works fine, but the RSI section freezes up. |
| L2 | March 22 | Group | Pass | Major | The program was passing into an incorrect variable. | Yes | This was causing the program to look at the wrong variable. |
| M1 | March 22 | Group | Fail | Major | Program is getting stuck on the cryptocurrency API call, not returning any values, freezing the program. | No | Program functions if the user avoids cryptocurrency, or else it crashes. |
| M2 | March 22 | Group | Pass | Major | Program makes the cryptocurrency API call properly. | Yes | The program was reinitializing the variable being used. |
| O1 | March 29 | Group | Fail | Trivial | When asked if the user wants to start monitoring, you can enter anything starting with ‘y’ or ‘n’. | No | This doesn’t break the program or hinder usability, simply means if you type “yellow”, it still works. |
| Q1 | March 29 | Group | Fail | Major | The user can enter the short-term longer than the long-term, causing the values to be used incorrectly. | No | This doesn’t break the program, but it does break the SMA option in that instance of running. |
| Q2 | March 29 | Group | Pass | Major | The user can no longer enter a short-term longer than the long-term. | Yes | We added a simple if statement to prevent this, and the user is asked to try again. |

## 3.2 INTEGRATION TESTING

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect | Summary of Defect | Closed prior to Production Release? | Comments |
| B1 | March 15 | Group | Fail | Trivial | URLConnection will occasionally return as “Source not found”, but will eventually return valid values. | No | This delays the program, but does not seem to break it. |
| E1 | March 15 | Group | Fail | Critical | The API is returning “Invalid API call”, rendering the program useless, as it returns with absolutely no information. | No |  |
| E2 | March 15 | Group | Pass | Critical | The API now returns values, allowing the program to function better. However, the API now returns an invalid number. | No | The API call was formatted incorrectly. Fixing the format allowed a valid API call. |
| E3 | March 15 | Group | Pass | Critical | The API is returning values with an extra quotation mark. | Yes | We adjusted the parse statement to avoid this. |
| F1 | March 19 | Group | Fail | Minor | The API is pulling the entire page of results instead of a single result. | No | This does not crash the program, but it causes invalid output as it is using old data. |
| F2 | March 19 | Group | Pass | Minor | The API is now pulling only the relevant results. | Yes | The program runs properly, only using recent data. |
| I1 | March 19 | Group | Fail | Major | The program can be started without entering any input. | No | This leads the program to run essentially nothing, infinitely. |
| I2 | March 19 | Group | Pass | Major | The program now requires the user to enter some input, waiting for an “n” response before it can begin monitoring. | Yes | The program no longer has a way to jump into an infinite non-functional loop. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major revamp of code | | | | | | | |
| N1 | March 26 | Group | Fail | Major | The program will allow you to enter *10%*, but not a string. This crashes the program if it is attempted, but the rest of the program is perfectly viable. | No | We could not find a resolution to this problem, as we could either stop non-integer values, or have an exception thrown. |
| P1 | March 29 | Group | Fail | Critical | The API can sometimes (rarely) return a 503 error, which throws an exception that crashes the program. | No | After running the program time after time after time, we’ve only seen this error twice. |

## 3.3 SYSTEM TESTING

**3.3.1 Functional requirements testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect | Summary of Defect | Closed prior to Production Release? | Comments |
| 3.2.1 | April 1 | Group | Pass | N/A | N/A | Yes | The user can select stocks and have them validated before working with them. |
| 3.2.2 | April 1 | Group | Pass | N/A | N/A | Yes | The user is offered a list of trading rules, and asked for the necessary values (also validated). |
| 3.2.3 | April 1 | Group | Pass | N/A | N/A | Yes | The program is pulling the API requested data. |
| 3.2.4 | April 1 | Group | Pass | N/A | N/A | Yes | The program is triggering properly. |
| 3.2.5 | April 1 | Group | Pass | N/A | N/A | Yes | The program displays a notification only when triggered. |

**3.3.2 Non-functional requirements testing**

* The program can run on multiple clients at the same time, with no change in the programs response time.
* However, the program does not require multiple clients to be running the program, it also functions properly with a single client.